

---

---

**Information technology — Coding of  
audio-visual objects —**

**Part 2:  
Visual**

*Technologies de l'information — Codage des objets audiovisuels —  
Partie 2: Codage visuel*



**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Withdrawn

© ISO/IEC 2001

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.ch](mailto:copyright@iso.ch)  
Web [www.iso.ch](http://www.iso.ch)

Printed in Switzerland

## Contents

1	Scope .....	1
2	Normative references .....	1
3	Terms and definitions .....	2
4	Abbreviations and symbols .....	12
4.1	Arithmetic operators .....	12
4.2	Logical operators .....	13
4.3	Relational operators .....	13
4.4	Bitwise operators .....	13
4.5	Conditional operators .....	13
4.6	Assignment .....	13
4.7	Mnemonics .....	14
4.8	Constants .....	14
5	Conventions .....	14
5.1	Method of describing bitstream syntax .....	14
5.2	Definition of functions .....	15
5.2.1	Definition of next_bits() function .....	15
5.2.2	Definition of bytealigned() function .....	15
5.2.3	Definition of nextbits_bytealigned() function .....	15
5.2.4	Definition of next_start_code() function .....	16
5.2.5	Definition of next_resync_marker() function .....	16
5.2.6	Definition of transparent_mb() function .....	16
5.2.7	Definition of transparent_block() function .....	16
5.2.8	Definition of byte_align_for_upstream() function .....	16
5.3	Reserved, forbidden and marker_bit .....	16
5.4	Arithmetic precision .....	17
6	Visual bitstream syntax and semantics .....	17
6.1	Structure of coded visual data .....	17
6.1.1	Visual object sequence .....	18
6.1.2	Visual object .....	18
6.1.3	Video object .....	18
6.1.4	Mesh object .....	24
6.1.5	FBA object .....	25
6.1.6	3D Mesh Object .....	29
6.2	Visual bitstream syntax .....	30
6.2.1	Start codes .....	30
6.2.2	Visual Object Sequence and Visual Object .....	34
6.2.3	Video Object Layer .....	35
6.2.4	Group of Video Object Plane .....	40
6.2.5	Video Object Plane and Video Plane with Short Header .....	40
6.2.6	Macroblock .....	53
6.2.7	Block .....	58
6.2.8	Still Texture Object .....	59
6.2.9	Mesh Object .....	73
6.2.10	FBA Object .....	75
6.2.11	3D Mesh Object .....	85
6.2.12	Upstream message .....	103
6.3	Visual bitstream semantics .....	104
6.3.1	Semantic rules for higher syntactic structures .....	104
6.3.2	Visual Object Sequence and Visual Object .....	104
6.3.3	Video Object Layer .....	109

6.3.4	Group of Video Object Plane .....	120
6.3.5	Video Object Plane and Video Plane with Short Header .....	120
6.3.6	Macroblock related .....	131
6.3.7	Block related .....	134
6.3.8	Still texture object.....	135
6.3.9	Mesh object .....	142
6.3.10	FBA object .....	144
6.3.11	3D Mesh Object.....	151
6.3.12	Upstream message .....	162
7	The visual decoding process .....	164
7.1	Video decoding process .....	165
7.2	Higher syntactic structures .....	166
7.3	VOP reconstruction .....	166
7.4	Texture decoding .....	166
7.4.1	Variable length decoding .....	167
7.4.2	Inverse scan .....	168
7.4.3	DC and AC prediction for intra macroblocks.....	169
7.4.4	Inverse quantisation .....	172
7.4.5	Inverse DCT .....	175
7.4.6	Upsampling of the Inverse DCT output for Reduced Resolution VOP.....	176
7.5	Shape decoding .....	177
7.5.1	Higher syntactic structures .....	177
7.5.2	Macroblock decoding.....	178
7.5.3	Arithmetic decoding.....	187
7.5.4	Spatial scalable binary shape decoding.....	189
7.5.5	Grayscale Shape Decoding .....	198
7.5.6	Multiple Auxiliary Component Decoding .....	201
7.6	Motion compensation decoding.....	201
7.6.1	Padding process.....	201
7.6.2	Sample interpolation for non-integer motion vectors.....	205
7.6.3	General motion vector decoding process.....	207
7.6.4	Unrestricted motion compensation .....	209
7.6.5	Vector decoding processing and motion-compensation in progressive P- and S(GMC)-VOP .....	210
7.6.6	Overlapped motion compensation.....	212
7.6.7	Temporal prediction structure .....	213
7.6.8	Vector decoding process of non-scalable progressive B-VOPs .....	214
7.6.9	Motion compensation in non-scalable progressive B-VOPs .....	214
7.6.10	Motion Compensation Decoding of Reduced Resolution VOP .....	219
7.7	Interlaced video decoding .....	224
7.7.1	Field DCT and DC and AC Prediction .....	224
7.7.2	Motion compensation.....	225
7.8	Sprite decoding.....	234
7.8.1	Higher syntactic structures .....	234
7.8.2	Sprite Reconstruction .....	235
7.8.3	Low-latency sprite reconstruction .....	235
7.8.4	Sprite reference point decoding.....	236
7.8.5	Warping.....	237
7.8.6	Sample reconstruction .....	239
7.8.7	GMC decoding.....	240
7.9	Generalized scalable decoding .....	241
7.9.1	Temporal scalability .....	241
7.9.2	Spatial scalability.....	246
7.10	Still texture object decoding.....	251
7.10.1	Decoding of the DC subband .....	251
7.10.2	ZeroTree Decoding of the Higher Bands.....	252
7.10.3	Inverse Quantisation .....	257
7.10.4	Still Texture Error Resilience.....	265
7.10.5	Wavelet Tiling.....	268
7.10.6	Scalable binary shape object decoding .....	270

7.11	Mesh object decoding .....	276
7.11.1	Mesh geometry decoding .....	276
7.11.2	Decoding of mesh motion vectors .....	279
7.12	FBA object decoding .....	281
7.12.1	Frame based face object decoding .....	281
7.12.2	DCT based face object decoding .....	282
7.12.3	Decoding of the viseme parameter fap 1 .....	284
7.12.4	Decoding of the viseme parameter fap 2 .....	284
7.12.5	Fap masking .....	285
7.12.6	Frame Based Body Decoding .....	285
7.12.7	DCT based body object decoding .....	286
7.13	3D Mesh Object Decoding .....	287
7.13.1	Start codes and bit stuffing .....	288
7.13.2	The Topological Surgery decoding process .....	288
7.13.3	The Forest Split decoding process .....	291
7.13.4	Header decoder .....	292
7.13.5	partition type .....	293
7.13.6	Vertex Graph Decoder .....	294
7.13.7	Triangle Tree Decoder .....	298
7.13.8	Triangle Data Decoder .....	299
7.13.9	Forest Split decoder .....	303
7.13.10	Arithmetic decoder .....	309
7.14	NEWPREP mode decoding .....	314
7.14.1	Decoder Definition .....	314
7.14.2	Upstream message .....	314
7.15	Output of the decoding process .....	314
7.15.1	Video data .....	315
7.15.2	2D Mesh data .....	315
7.15.3	Face animation parameter data .....	315
8	Visual-Systems Composition Issues .....	315
8.1	Temporal Scalability Composition .....	315
8.2	Sprite Composition .....	316
8.3	Mesh Object Composition .....	317
8.4	Spatial Scalability composition .....	318
9	Profiles and Levels .....	318
9.1	Visual Object Types .....	318
9.2	Visual Profiles .....	321
9.3	Visual Profiles@Levels .....	322
9.3.1	Natural Visual .....	322
9.3.2	Synthetic Visual .....	322
9.3.3	Synthetic/Natural Hybrid Visual .....	324
Annex A	(normative) Coding transforms .....	326
A.1	Discrete cosine transform for video texture .....	326
A.2	Discrete wavelet transform for still texture .....	327
A.2.1	Adding the mean .....	327
A.2.2	Wavelet filter .....	327
A.2.3	Symmetric extension .....	328
A.2.4	Decomposition level .....	329
A.2.5	Shape adaptive wavelet filtering and symmetric extension .....	329
A.3	Shape-Adaptive DCT (SA-DCT) .....	330
A.3.1	Definition of Forward SA-DCT .....	330
A.3.2	Definition of Inverse SA-DCT .....	332
A.4	SA-DCT with DC Separation and $\Delta$ DC Correction ( $\Delta$ DC-SA-DCT) .....	333
A.4.1	Definition of Forward $\Delta$ DC-SA-DCT .....	334
A.4.2	Definition of Inverse $\Delta$ DC-SA-DCT .....	334
Annex B	(normative) Variable length codes and arithmetic decoding .....	336
B.1	Variable length codes .....	336
B.1.1	Macroblock type .....	336

B.1.2	Macroblock pattern.....	338
B.1.3	Motion vector .....	340
B.1.4	DCT coefficients.....	342
B.1.5	Shape Coding.....	352
B.1.6	Sprite Coding .....	357
B.1.7	DCT based facial object decoding .....	358
B.1.8	Shape decoding for still texture object .....	367
B.2	Arithmetic Decoding.....	368
B.2.1	Arithmetic decoding for still texture object .....	368
B.2.2	Arithmetic decoding for shape decoding.....	371
B.2.3	FBA Object Decoding.....	374
Annex C	(normative) Face and body object decoding tables and definitions.....	376
Annex D	(normative) Video buffering verifier.....	409
D.1	Introduction .....	409
D.2	Video Rate Buffer Model Definition.....	409
D.3	Comparison between ISO/IEC 14496-2 VBV and the ISO/IEC 13818-2 VBV (informative).....	412
D.4	Video Complexity Model Definition.....	413
D.5	Video Reference Memory Model Definition.....	415
D.6	Interaction between VBV, VCV and VMV (informative).....	416
D.7	Video Presentation Model Definition (informative) .....	416
Annex E	(informative) Features supported by the algorithm .....	418
E.1	Error resilience.....	418
E.1.1	Resynchronization.....	418
E.1.2	Data Partitioning .....	419
E.1.3	Reversible VLC.....	419
E.1.4	Decoder Operation.....	420
E.1.5	Adaptive Intra Refresh (AIR) Method .....	423
E.1.6	NEWPRED.....	425
E.2	Complexity Estimation .....	427
E.3	Resynchronization in Case of Unknown Video Header Format.....	427
Annex F	(informative) Preprocessing and postprocessing.....	428
F.1	VOP Generation Tools: Automatic and Semi-automatic Segmentations .....	428
F.1.1	Automatic Segmentation .....	428
F.1.2	Semi-automatic Segmentation .....	438
F.1.3	References.....	446
F.2	Bounding Rectangle of VOP Formation .....	447
F.3	Postprocessing for Coding Noise Reduction .....	448
F.3.1	Deblocking filter.....	448
F.3.2	Deringing filter.....	450
F.3.3	Further issues .....	452
F.4	Chrominance Decimation and Interpolation Filtering for Interlaced Object Coding .....	452
Annex G	(normative) Profile and level indication and restrictions .....	454
Annex H	(informative) Patent statements .....	457
H.1	Patent statements for ISO/IEC 14496 Version 1 .....	457
H.2	Patent statements for the extensions provided in ISO/IEC 14496 Version 2 .....	458
Annex I	(informative) Encoder Complexity Reduction Based on Intelligent Pre-Quantisation .....	460
I.1	Introduction .....	460
I.2	Feature Selection and Pre-quantisation.....	460
I.3	Model Verification and Threshold Setting.....	462
I.3.1	H.263 Quantiser.....	462
I.3.2	MPEG-4 Quantiser .....	462
Annex J	(normative) View dependent object scalability .....	464
J.1	Introduction .....	464
J.2	Decoding Process of a View-Dependent Object.....	464
J.2.1	General Decoding Scheme .....	464
J.2.2	Computation of the View-Dependent Scalability parameters .....	466

J.2.3	VD mask computation .....	468
J.2.4	Differential mask computation .....	469
J.2.5	DCT coefficients decoding .....	469
J.2.6	Texture update .....	469
J.2.7	IDCT .....	470
<b>Annex K</b>	<b>(normative) Decoder Configuration Information .....</b>	<b>471</b>
K.1	Introduction .....	471
K.2	Description of the set up of a visual decoder (informative) .....	471
K.2.1	Processing of decoder configuration information .....	472
K.3	Specification of decoder configuration information .....	473
K.3.1	VideoObject .....	473
K.3.2	StillTextureObject .....	473
K.3.3	MeshObject .....	474
K.3.4	FaceObject .....	474
K.3.5	3DMeshObject .....	474
<b>Annex L</b>	<b>(informative) Rate control .....</b>	<b>475</b>
L.1	Frame Rate Control .....	475
L.1.1	Introduction .....	475
L.1.2	Description .....	475
L.1.3	Summary .....	479
L.2	Multiple Video Object Rate Control .....	479
L.2.1	Initialization .....	479
L.2.2	Quantisation Level Calculation for I-frame and first P-frame .....	479
L.2.3	Update Rate-Distortion Model .....	482
L.2.4	Post-Frameskip Control .....	482
L.3	Macroblock Rate Control .....	484
L.3.1	Rate-Distortion Model .....	484
L.3.2	Target Number of Bits for Each Macroblock .....	485
L.3.3	Macroblock Rate Control .....	485
<b>Annex M</b>	<b>(informative) Binary shape coding .....</b>	<b>487</b>
M.1	Introduction .....	487
M.2	Context-Based Arithmetic Shape Coding .....	487
M.2.1	Intra Mode .....	488
M.2.2	Inter Mode .....	488
M.3	Texture Coding of Boundary Blocks .....	489
M.4	Encoder Architecture .....	489
M.5	Encoding Guidelines .....	490
M.5.1	Lossy Shape Coding .....	491
M.5.2	Coding Mode Selection .....	491
M.6	Conclusions .....	491
M.7	References .....	492
<b>Annex N</b>	<b>(normative) Visual profiles@levels .....</b>	<b>493</b>
<b>Annex O</b>	<b>(informative) 3D Mesh Coding .....</b>	<b>497</b>
O.1	Introduction .....	497
O.2	Topological Surgery Representation .....	497
O.2.1	Simple Polygon Representation .....	498
O.2.2	Vertex Graph representation .....	499
O.3	Encoding guidelines for 3D Mesh Coding .....	500
O.3.1	Topological Surgery Encoding .....	500
O.3.2	Support for non-manifolds and Non-orientable manifolds .....	501
O.3.3	Support for Error Resilience .....	503
O.4	Encoder considerations for efficient compression of Vertex Properties .....	507
O.5	Progressive Forest Split Representation .....	508
O.5.1	Encoding the Forest .....	508
O.5.2	Support for meshes with polygonal faces .....	509
O.5.3	Method for generating of a PFS Representation of a Triangular 3D Mesh .....	509
O.5.4	Topological Tests .....	510

O.5.5 Geometric Tests.....512

O.6 Complexity estimation for Computational Graceful Degradation .....512

O.7 QoS for SNHC through upstream.....514

O.8 References.....516

Bibliography .....517

Withdrawn



## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 14496 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14496-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 14496-2:1999), which has been technically revised. It incorporates Cor.1:2000, Cor.2:2001 and Amd.1:2000.

ISO/IEC 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

- *Part 1: Systems*
- *Part 2: Visual*
- *Part 3: Audio*
- *Part 4: Conformance testing*
- *Part 5: Reference software*
- *Part 6: Delivery Multimedia Integration Framework (DMIF)*
- *Part 7: Optimized software for MPEG-4 visual tools*
- *Part 8: Carriage of MPEG-4 contents over IP networks*

Annexes A to D, G, J, K and N form a normative part of this part of ISO/IEC 14496. Annexes E, F, H, I, L, M and O are for information only.

## Introduction

### Purpose

This part of ISO/IEC 14496 was developed in response to the growing need for a coding method that can facilitate access to visual objects in natural and synthetic moving pictures and associated natural or synthetic sound for various applications such as digital storage media, internet, various forms of wired or wireless communication etc. The use of ISO/IEC 14496 means that motion video can be manipulated as a form of computer data and can be stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcast channels.

### Application

The applications of ISO/IEC 14496 cover, but are not limited to, such areas as listed below:

IMM	Internet Multimedia
IVG	Interactive Video Games
IPC	Interpersonal Communications (videoconferencing, videophone, etc.)
ISM	Interactive Storage Media (optical disks, etc.)
MMM	Multimedia Mailing
NDB	Networked Database Services (via ATM, etc.)
RES	Remote Emergency Systems
RVS	Remote Video Surveillance
WMM	Wireless Multimedia
Multimedia	

### Profiles and levels

ISO/IEC 14496 is intended to be generic in the sense that it serves a wide range of applications, bitrates, resolutions, qualities and services. Furthermore, it allows a number of modes of coding of both natural and synthetic video in a manner facilitating access to individual objects in images or video, referred to as content based access. Applications should cover, among other things, digital storage media, content based image and video databases, internet video, interpersonal video communications, wireless video etc. In the course of creating ISO/IEC 14496, various requirements from typical applications have been considered, necessary algorithmic elements have been developed, and they have been integrated into a single syntax. Hence ISO/IEC 14496 will facilitate the bitstream interchange among different applications.

This part of ISO/IEC 14496 includes one or more complete decoding algorithms as well as a set of decoding tools. Moreover, the various tools of this part of ISO/IEC 14496 as well as that derived from ISO/IEC 13818-2:1996 can be combined to form other decoding algorithms. Considering the practicality of implementing the full syntax of this part of ISO/IEC 14496, however, a limited number of subsets of the syntax are also stipulated by means of “profile” and “level”.

A “profile” is a defined subset of the entire bitstream syntax that is defined by this part of ISO/IEC 14496. Within the bounds imposed by the syntax of a given profile it is still possible to require a very large variation in the performance of encoders and decoders depending upon the values taken by parameters in the bitstream.

In order to deal with this problem “levels” are defined within each profile. A level is a defined set of constraints imposed on parameters in the bitstream. These constraints may be simple limits on numbers. Alternatively they may take the form of constraints on arithmetic combinations of the parameters.

## Object based coding syntax

### Video object

A *video object* in a scene is an entity that a user is allowed to access (seek, browse) and manipulate (cut and paste). The instances of video objects at a given time are called *video object planes* (VOPs). The encoding process generates a coded representation of a VOP as well as composition information necessary for display. Further, at the decoder, a user may interact with and modify the composition process as needed.

The full syntax allows coding of rectangular as well as arbitrarily shaped video objects in a scene. Furthermore, the syntax supports both non-scalable coding and scalable coding. Thus it becomes possible to handle normal scalabilities as well as object based scalabilities. The scalability syntax enables the reconstruction of useful video from pieces of a total bitstream. This is achieved by structuring the total bitstream in two or more layers, starting from a standalone base layer and adding a number of enhancement layers. The base layer can be coded using a non-scalable syntax, or in the case of picture based coding, even using a syntax of a different video coding standard.

To ensure the ability to access individual objects, it is necessary to achieve a coded representation of its shape. A natural video object consists of a sequence of 2D representations (at different points in time) referred to here as VOPs. For efficient coding of VOPs, both temporal redundancies as well as spatial redundancies are exploited. Thus a coded representation of a VOP includes representation of its shape, its motion and its texture.

### FBA object

The FBA object is a collection of nodes in a scene graph which are animated by the FBA (Face and Body Animation) object bitstream. The FBA object is controlled by two separate bitstreams. The first bitstream, called BIFS, contains instances of Body Definition Parameters (BDPs) in addition to Facial Definition Parameters (FDPs), and the second bitstream, FBA bitstream, contains Body Animation Parameters (BAPs) together with Facial Animation Parameters (FAPs).

A 3D (or 2D) *face object* is a representation of the human face that is structured for portraying the visual manifestations of speech and facial expressions adequate to achieve visual speech intelligibility and the recognition of the mood of the speaker. A face object is animated by a stream of *face animation parameters* (FAP) encoded for low-bandwidth transmission in broadcast (one-to-many) or dedicated interactive (point-to-point) communications. The FAPs manipulate key feature control points in a mesh model of the face to produce animated visemes for the mouth (lips, tongue, teeth), as well as animation of the head and facial features like the eyes. FAPs are quantized with careful consideration for the limited movements of facial features, and then prediction errors are calculated and coded arithmetically. The remote manipulation of a face model in a terminal with FAPs can accomplish lifelike visual scenes of the speaker in real-time without sending pictorial or video details of face imagery every frame.

A simple streaming connection can be made to a decoding terminal that animates a default face model. A more complex session can initialize a custom face in a more capable terminal by downloading *face definition parameters* (FDP) from the encoder. Thus specific background images, facial textures, and head geometry can be portrayed. The composition of specific backgrounds, face 2D/3D meshes, texture attribution of the mesh, etc. is described in ISO/IEC 14496-1:1999. The FAP stream for a given user can be generated at the user's terminal from video/audio, or from text-to-speech. FAPs can be encoded at bitrates up to 2-3kbit/s at necessary speech rates. Optional temporal DCT coding provides further compression efficiency in exchange for delay. Using the facilities of ISO/IEC 14496-1:1999, a composition of the animated face model and synchronized, coded speech audio (low-bitrate speech coder or text-to-speech) can provide an integrated low-bandwidth audio/visual speaker for broadcast applications or interactive conversation.

Limited scalability is supported. Face animation achieves its efficiency by employing very concise motion animation controls in the channel, while relying on a suitably equipped terminal for rendering of moving 2D/3D faces with non-normative models held in local memory. Models stored and updated for rendering in the terminal can be simple or complex. To support speech intelligibility, the normative specification of FAPs intends for their selective or complete use as signaled by the encoder. A masking scheme provides for selective transmission of FAPs according to what

parts of the face are naturally active from moment to moment. A further control in the FAP stream allows face animation to be suspended while leaving face features in the terminal in a defined quiescent state for higher overall efficiency during multi-point connections.

A body model is a representation of a virtual human or human-like character that allows portraying body movements adequate to achieve nonverbal communication and general actions. A body model is animated by a stream of *body animation parameters* (BAP) encoded for low-bitrate transmission in broadcast and dedicated interactive communications. The BAPs manipulate independent degrees of freedom in the skeleton model of the body to produce animation of the body parts. The BAPs are quantized considering the joint limitations, and prediction errors are calculated and coded arithmetically. Similar to the face, the remote manipulation of a body model in a terminal with BAPs can accomplish lifelike visual scenes of the body in real-time without sending pictorial and video details of the body every frame.

The BAPs, if correctly interpreted, will produce reasonably similar high level results in terms of body posture and animation on different body models, also without the need to initialize or calibrate the model. The BDP set defines the set of parameters to transform the default body to a customized body optionally with its body surface, body dimensions, and texture.

The *body definition parameters* (BDP) allow the encoder to replace the local model of a more capable terminal. BDP parameters include body geometry, calibration of body parts, degrees of freedom, and optionally deformation information.

The FBA Animation specification is defined in ISO/IEC 14496-1:1999/Amd.1:2000 and this part of ISO/IEC 14496. This clause is intended to facilitate finding various parts of specification. As a rule of thumb, FAP and BAP specification is found in the part 2, and FDP and BDP specification in the part 1. However, this is not a strict rule. For an overview of FAPs/BAPs and their interpretation, read subclauses "6.1.5.2 Facial animation parameter set", "6.1.5.3 Facial animation parameter units", "6.1.5.4 Description of a neutral face" as well as the Table C-1. The viseme parameter is documented in subclause "7.12.3 Decoding of the viseme parameter fap 1" and the Table C-5 in annex C. The expression parameter is documented in subclause "7.12.4 Decoding of the expression parameter fap 2" and the Table C-3. FBA bitstream syntax is found in subclauses "6.2.10 FBA Object", semantics in "6.3.10 FBA Object", and subclause "7.12 FBA object decoding" explains in more detail the FAP/BAP decoding process. FAP/BAP masking and interpolation is explained in subclauses "6.3.11.1 FBA Object Plane", "7.12.1.1 Decoding of FBA", "7.12.5 FBA masking". The FIT interpolation scheme is documented in subclause "7.2.5.3.2.4 FIT" of ISO/IEC 14496-1:1999. The FDRs and BDPs and their interpretation are documented in subclause "7.2.5.3.2.6 FDP" of ISO/IEC 14496-1:1999. In particular, the FDP feature points are documented in Figure C-1. Details on body models are documented in annex C.

## Mesh object

A 2D *mesh object* is a representation of a 2D deformable geometric shape, with which synthetic video objects may be created during a composition process at the decoder, by spatially piece-wise warping of existing video object planes or still texture objects. The instances of mesh objects at a given time are called *mesh object planes* (mops). The geometry of mesh object planes is coded losslessly. Temporally and spatially predictive techniques and variable length coding are used to compress 2D mesh geometry. The coded representation of a 2D mesh object includes representation of its geometry and motion.

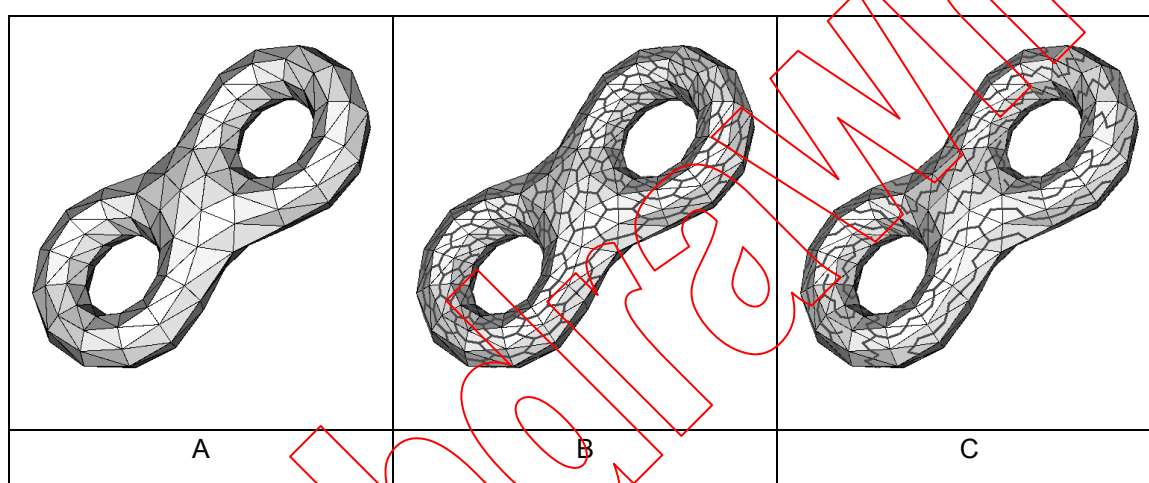
## 3D Mesh Object

The 3D Mesh Object is a 3D polygonal model that can be represented as an IndexedFaceSet or Hierarchical 3D Mesh node in BIFS. It is defined by the position of its vertices (geometry), by the association between each face and its sustaining vertices (connectivity), and optionally by colours, normals, and texture coordinates (properties). Properties do not affect the 3D geometry, but influence the way the model is shaded. 3D mesh coding (3DMC) addresses the efficient coding of 3D mesh object. It comprises a basic method and several options. The basic 3DMC method operates on manifold model and features incremental representation of single resolution 3D model. The model may be triangular or polygonal – the latter are triangulated for coding purposes and are fully recovered in the decoder. Options include: (a) support for computational graceful degradation control; (b) support for non-manifold model; (c) support for error resilience; and (d) quality scalability via hierarchical transmission of levels of detail with implicit support for smooth transition between consecutive levels. The compression of application-specific geometry streams (Face Animation Parameters) and generalized animation parameters (BIFS Anim) are currently addressed elsewhere in this part of ISO/IEC 14496.

In 3DMC, the compression of the connectivity of the 3D mesh (e.g. how edges, faces, and vertices relate) is lossless, whereas the compression of the other attributes (such as vertex coordinates, normals, colours, and texture coordinates) may be lossy.

### Single Resolution Mode

The incremental representation of a single resolution 3D model is based on the Topological Surgery scheme. For manifold triangular 3D meshes, the Topological Surgery representation decomposes the connectivity of each *connected component* into a *simple polygon* and a *vertex graph*. All the triangular faces of the 3D mesh are connected in the simple polygon forming a *triangle tree*, which is a spanning tree in the dual graph of the 3D mesh. Figure V2 - 1 shows an example of a triangular 3D mesh, its dual graph, and a triangle tree. The vertex graph identifies which pairs of boundary edges of the simple polygon are associated with each other to reconstruct the connectivity of the 3D mesh. The triangle tree does not fully describe the triangulation of the simple polygon. The missing information is recorded as a *marching edge*.



**Figure V2 - 1 -- A triangular 3D mesh (A), its dual graph (B), and a triangle tree (C)**

For manifold 3D meshes, the connectivity is represented in a similar fashion. The polygonal faces of the 3D mesh are connected in a simple polygon forming a *face tree*. The faces are triangulated, and which edges of the resulting triangular 3D mesh are edges of the original 3D mesh is recorded as a sequence of *polygon\_edge* bits. The face tree is also a spanning tree in the dual graph of the 3D mesh, and the vertex graph is always composed of edges of the original 3D mesh.

The vertex coordinates and optional properties of the 3D mesh (normals, colours, and texture coordinates) are quantized, predicted as a function of decoded ancestors with respect to the order of traversal, and the errors are entropy encoded.

### Incremental Representation

When a 3D mesh is downloaded over networks with limited bandwidth (e.g. PSTN), it may be desired to begin decoding and rendering the 3D mesh before it has all been received. Moreover, content providers may wish to control such incremental representation to present the most important data first. The basic 3DMC method supports this by interleaving the data such that each triangle may be reconstructed as it is received. Incremental representation is also facilitated by the options of hierarchical transmission for quality scalability and partitioning for error resilience.

### Hierarchical Mode

An example of a 3D mesh represented in hierarchical mode is illustrated in Figure V2 - 2. The hierarchical mode allows the decoder to show progressively better approximations of the model as data are received. The hierarchical 3D mesh decomposition can also be organized in the decoder as layered detail, and view-dependent expansion of this detail can be subsequently accomplished during a viewer's interaction with the 3D model. Downloadable



scalability of 3D model allows decoders with widely varied rendering performance to use the content, without necessarily making repeated requests to the encoder for specific versions of the content and without the latencies of updating view-dependent model from the encoder. This quality scalability of 3D meshes is complementary to scaleable still texture and supports bitstream scalability and downloading of quality hierarchies of hybrid imagery to the decoder.

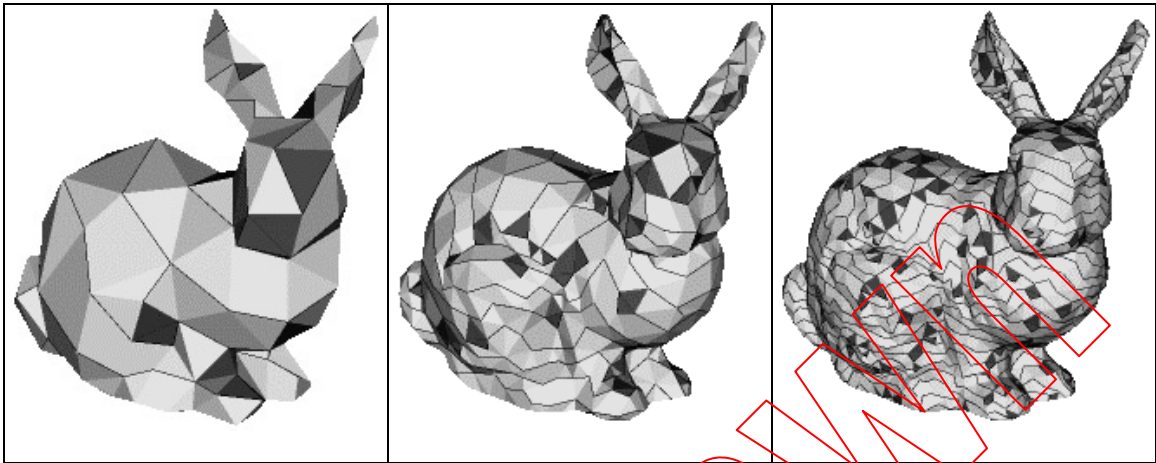


Figure V2 - 2 -- A hierarchy of levels of detail

The hierarchical representation of a 3D model is based on the Progressive Forest Split scheme. In the Progressive Forest Split representation, a manifold 3D mesh is represented as a *base 3D mesh* followed by a sequence of *forest split* operations. The base 3D mesh is encoded as a single resolution 3D mesh using the Topological Surgery scheme. Each forest split operation is composed of a *forest* in the graph of the current 3D mesh, and a *sequence of simple polygons*. Figure V2 - 3 illustrates the method. To prevent visual artifacts which may occur while switching from a level of detail to the next one, the data structure supports smooth transition between consecutive levels of detail in the form of linear interpolation of vertex positions.

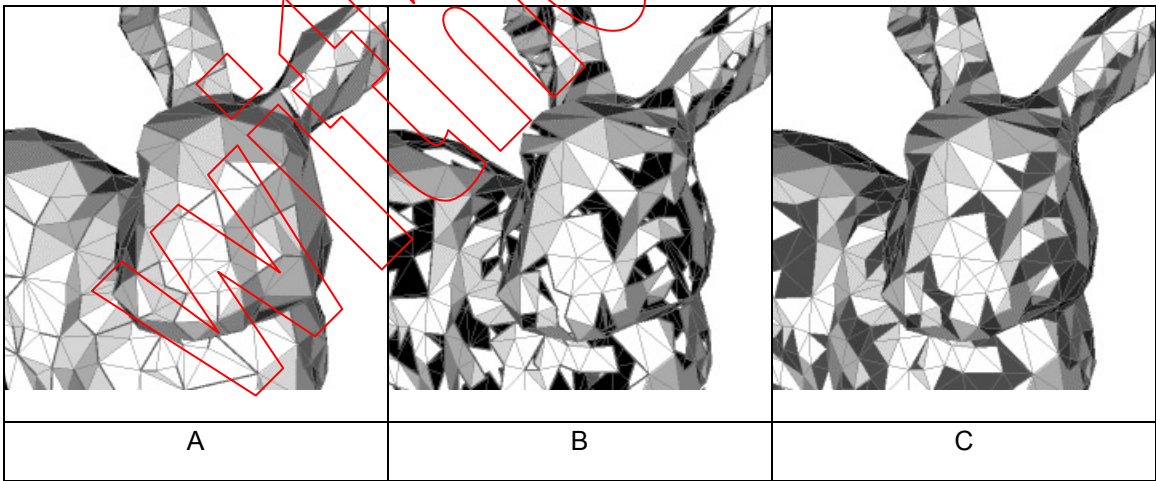


Figure V2 - 3 -- The Forest Split operation. A 3D mesh with a forest of edges marked (A). The tree loops resulting from cutting the 3D mesh through the forest edges (B). Each tree loop is filled by a simple polygon composed of polygonal faces (C).

Error Resilience for 3D mesh object

If the 3D mesh is partitioned into independent parts, it may be possible to perform more efficient data transmission in an error-prone environment, e.g., an IP network or datacasting service in a broadcast TV network. It must be possible to resynchronize after a channel error, and continue data transmission and rendering from that point instead of starting over from scratch. Even with the presence of channel errors, the decoder can start decoding and rendering from the next partition that is received intact from the channel.

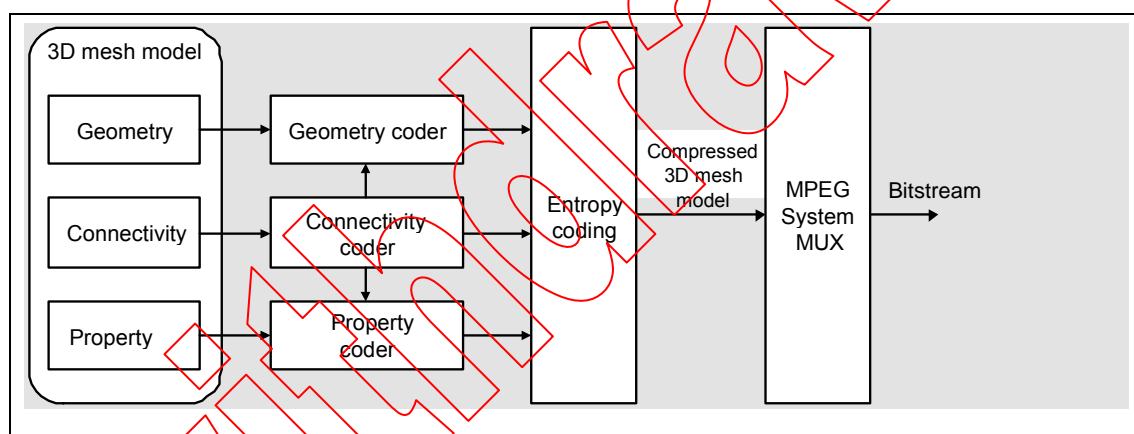
Flexible partitioning methods can be used to organize the data, such that it fits the underlying network packet structure more closely, and overhead is reduced to the minimum. To allow flexible partitioning, several connected components may be merged into one partition, where as a large connected component may be divided into several independent partitions. Merging and dividing of connected components using different partition types can be done at any point in the 3D mesh object.

### Stitching for Non-Manifold and Non-Orientable Meshes

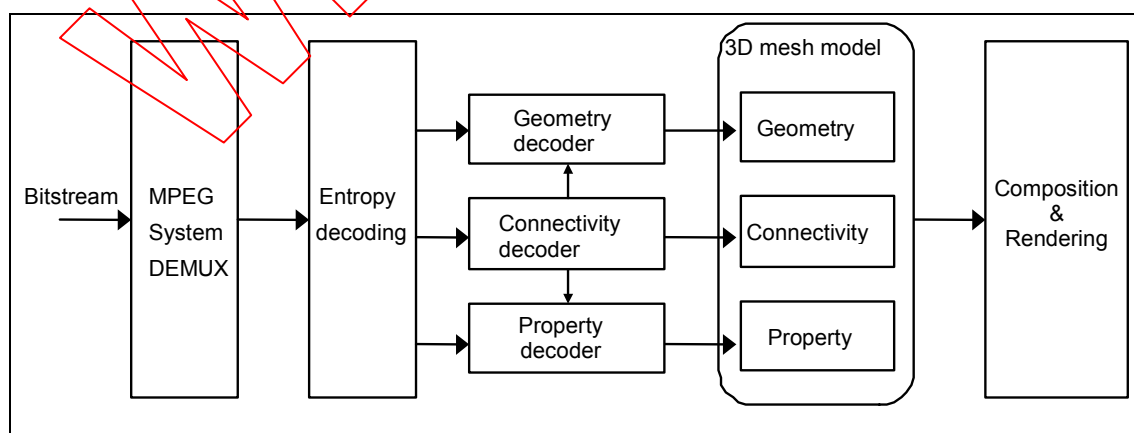
The connectivity of a non-manifold and non-orientable 3D mesh is represented as a manifold 3D mesh and a sequence of *stitches*. Each stitch describes how to identify one or more pairs of vertices along two linear paths of edges, and each one of these paths is contained in one of the vertex graphs that span the connected components of the manifold 3D mesh.

### Encoder and Decoder Block Diagrams

High level block diagrams of a general 3D polygonal model encoder and decoder are shown in Figure V2 - 4. They consist of a 3D mesh connectivity (de)coder, geometry (de)coder, property (de)coder, and entropy (de)coding blocks. Connectivity, vertex position, and property information are extracted from 3D mesh model described in VRML or MPEG-4 BIFS format. The connectivity (de)coder is used for an efficient representation of the association between each face and its sustaining vertices. The geometry (de)coder is used for a lossy or lossless compression of vertex coordinates. The property (de)coder is used for a lossy or lossless compression of colour, normal, and texture coordinate data.



(a) 3D mesh encoder



(b) 3D mesh decoder

**Figure V2 - 4 -- General block diagram of the 3D mesh compress**

## Overview of the object based nonscalable syntax

The coded representation defined in the non-scalable syntax achieves a high compression ratio while preserving good image quality. Further, when access to individual objects is desired, the shape of objects also needs to be coded, and depending on the bandwidth available, the shape information can be coded in a lossy or lossless fashion.

The compression algorithm employed for texture data is not lossless as the exact sample values are not preserved during coding. Obtaining good image quality at the bitrates of interest demands very high compression, which is not achievable with intra coding alone. The need for random access, however, is best satisfied with pure intra coding. The choice of the techniques is based on the need to balance a high image quality and compression ratio with the requirement to make random access to the coded bitstream.

A number of techniques are used to achieve high compression. The algorithm first uses block-based motion compensation to reduce the temporal redundancy. Motion compensation is used both for causal prediction of the current VOP from a previous VOP, and for non-causal, interpolative prediction from past and future VOPs. Motion vectors are defined for each 16-sample by 16-line region of a VOP or 8-sample by 8-line region of a VOP as required. The prediction error, is further compressed using the discrete cosine transform (DCT) to remove spatial correlation before it is quantised in an irreversible process that discards the less important information. Finally, the shape information, motion vectors and the quantised DCT information, are encoded using variable length codes.

## Temporal processing

Because of the conflicting requirements of random access to and highly efficient compression, three main VOP types are defined. Intra coded VOPs (I-VOPs) are coded without reference to other pictures. They provide access points to the coded sequence where decoding can begin, but are coded with only moderate compression. Predictive coded VOPs (P-VOPs) are coded more efficiently using motion compensated prediction from a past intra or predictive coded VOPs and are generally used as a reference for further prediction. Bidirectionally-predictive coded VOPs (B-VOPs) provide the highest degree of compression but require both past and future reference VOPs for motion compensation. Bidirectionally-predictive coded VOPs are never used as references for prediction (except in the case that the resulting VOP is used as a reference for scalable enhancement layer). The organisation of the three VOP types in a sequence is very flexible. The choice is left to the encoder and will depend on the requirements of the application.

## Coding of Shapes

In natural video scenes, VOPs are generated by segmentation of the scene according to some semantic meaning. For such scenes, the shape information is thus binary (binary shape). Shape information is also referred to as alpha plane. The binary alpha plane is coded on a macroblock basis by a coder which uses the context information, motion compensation and arithmetic coding.

For coding of shape of a VOP, a bounding rectangle is first created and is extended to multiples of  $16 \times 16$  blocks with extended alpha samples set to zero. Shape coding is then initiated on a  $16 \times 16$  block basis; these blocks are also referred to as binary alpha blocks.

## Motion representation - macroblocks

The choice of  $16 \times 16$  blocks (referred to as macroblocks) for the motion-compensation unit is a result of the trade-off between the coding gain provided by using motion information and the overhead needed to represent it. Each macroblock can further be subdivided to  $8 \times 8$  blocks for motion estimation and compensation depending on the overhead that can be afforded. In order to encode the highly active scene with higher VOP rate, a Reduced Resolution VOP tool is provided. When this tool is used, the size of the macroblock used for motion compensation decoding is  $32 \times 32$  pixels and the size of block is  $16 \times 16$  pixels.

Depending on the type of the macroblock, motion vector information and other side information is encoded with the compressed prediction error in each macroblock. The motion vectors are differenced with respect to a prediction value and coded using variable length codes. The maximum length of the motion vectors allowed is decided at the encoder. It is the responsibility of the encoder to calculate appropriate motion vectors. The specification does not specify how this should be done.



## Spatial redundancy reduction

Both source VOPs and prediction errors VOPs have significant spatial redundancy. This part of ISO/IEC 14496 uses a block-based DCT method with optional visually weighted quantisation, and run-length coding. After motion compensated prediction or interpolation, the resulting prediction error is split into  $8 \times 8$  blocks. These are transformed into the DCT domain where they can be weighted before being quantised. After quantisation many of the DCT coefficients are zero in value and so two-dimensional run-length and variable length coding is used to encode the remaining DCT coefficients efficiently.

## Chrominance formats

This part of ISO/IEC 14496 currently supports the 4:2:0 chrominance format.

## Pixel depth

This part of ISO/IEC 14496 supports pixel depths between 4 and 12 bits in luminance and chrominance planes.

## Generalized scalability

The scalability tools in this part of ISO/IEC 14496 are designed to support applications beyond that supported by single layer video. The major applications of scalability include internet video, wireless video, multi-quality video services, video database browsing etc. In some of these applications, either normal scalabilities on picture basis such as those in ISO/IEC 13818-2:1996 may be employed or object based scalabilities may be necessary; both categories of scalability are enabled by this part of ISO/IEC 14496.

Although a simple solution to scalable video is the simulcast technique that is based on transmission/storage of multiple independently coded reproductions of video, a more efficient alternative is scalable video coding, in which the bandwidth allocated to a given reproduction of video can be partially re-utilised in coding of the next reproduction of video. In scalable video coding, it is assumed that given a coded bitstream, decoders of various complexities can decode and display appropriate reproductions of coded video. A scalable video encoder is likely to have increased complexity when compared to a single layer encoder. However, this part of ISO/IEC 14496 provides several different forms of scalabilities that address non-overlapping applications with corresponding complexities.

The basic scalability tools offered are temporal scalability and spatial scalability. Moreover, combinations of these basic scalability tools are also supported and are referred to as hybrid scalability. In the case of basic scalability, two layers of video referred to as the lower layer and the enhancement layer are allowed, whereas in hybrid scalability up to four layers are supported.

### Object based Temporal scalability

Temporal scalability is a tool intended for use in a range of diverse video applications from video databases, internet video, wireless video and multiview/stereoscopic coding of video. Furthermore, it may also provide a migration path from current lower temporal resolution video systems to higher temporal resolution systems of the future.

Temporal scalability involves partitioning of VOPs into layers, where the lower layer is coded by itself to provide the basic temporal rate and the enhancement layer is coded with temporal prediction with respect to the lower layer. These layers when decoded and temporally multiplexed yield full temporal resolution. The lower temporal resolution systems may only decode the lower layer to provide basic temporal resolution whereas enhanced systems of the future may support both layers. Furthermore, temporal scalability has use in bandwidth constrained networked applications where adaptation to frequent changes in allowed throughput are necessary. An additional advantage of temporal scalability is its ability to provide resilience to transmission errors as the more important data of the lower layer can be sent over a channel with better error performance, whereas the less critical enhancement layer can be sent over a channel with poor error performance. Object based temporal scalability can also be employed to allow graceful control of picture quality by controlling the temporal rate of each video object under the constraint of a given bit-budget.

## Object Spatial scalability

Spatial scalability is a tool intended for use in video applications involving multi quality video services, video database browsing, internet video and wireless video, i.e., video systems with the primary common feature that a minimum of two layers of spatial resolution are necessary. Spatial scalability involves generating two spatial resolution video layers from a single video source such that the lower layer is coded by itself to provide the basic spatial resolution and the enhancement layer employs the spatially interpolated lower layer and carries the full spatial resolution of the input video source.

Object based spatial scalability is composed of texture spatial scalability and shape spatial scalability. They can be used independently or together by its application. Binary shape spatial scalability is used for the applications that have 'binary shape only' mode as well as the applications of general object based spatial scalability.

An additional advantage of spatial scalability is its ability to provide resilience to transmission errors as the more important data of the lower layer can be sent over a channel with better error performance, whereas the less critical enhancement layer data can be sent over a channel with poor error performance. Further, it can also allow interoperability between various standards.

## Hybrid scalability

There are a number of applications where neither the temporal scalability nor the spatial scalability may offer the necessary flexibility and control. This may necessitate use of temporal and spatial scalability simultaneously and is referred to as the hybrid scalability. Among the applications of hybrid scalability are wireless video, internet video, multiviewpoint/stereoscopic coding etc.

## Error Resilience

This part of ISO/IEC 14496 provides error robustness and resilience to allow accessing of image or video information over a wide range of storage and transmission media. The error resilience tools developed for this part of ISO/IEC 14496 can be divided into three major categories. These categories include synchronization, data recovery, and error concealment. It should be noted that these categories are not unique to this part of ISO/IEC 14496, and have been used elsewhere in general research in this area. It is, however, the tools contained in these categories that are of interest, and where this part of ISO/IEC 14496 makes its contribution to the problem of error resilience.

# Information technology — Coding of audio-visual objects —

## Part 2: Visual

### 1 Scope

This part of ISO/IEC 14496 specifies the coded representation of picture information in the form of natural or synthetic visual objects like video sequences of rectangular or arbitrarily shaped pictures, moving 2D meshes, animated 3D face and body models and texture for synthetic objects. The coded representation allows for content based access for digital storage media, digital video communication and other applications. ISO/IEC 14496 specifies also the decoding process of the aforementioned coded representation. The representation supports constant bitrate transmission, variable bitrate transmission, robust transmission, content based random access (including normal random access), object based scalable decoding (including normal scalable decoding), object based bitstream editing, as well as special functions such as fast forward playback, fast reverse playback, slow motion, pause and still pictures. Synthetic objects and coding of special 2D/3D meshes, texture, and animation parameters are provided for use with downloadable models to exploit mixed media and the bandwidth improvement associated with remote manipulation of such models. ISO/IEC 14496 is intended to allow some level of interoperability with ISO/IEC 11172-2:1993, ISO/IEC 13818-2:1996 and ITU-T Recommendation H.263.

### 2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 14496. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 14496 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ITU-T Recommendation T.81 (1992)|ISO/IEC 10918-1:1994, *Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines*.

ISO/IEC 11172-1:1993, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 1: Systems*.

ISO/IEC 11172-2:1993, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 2: Video*.

ISO/IEC 11172-3:1993, *Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 3: Audio*.

ITU-T Recommendation H.222.0(2000)|ISO/IEC 13818-1:2000, *Information technology — Generic coding of moving pictures and associated audio information: Systems*.

ITU-T Recommendation H.262(2000)|ISO/IEC 13818-2:2000, *Information technology — Generic coding of moving pictures and associated audio information: Video*.

ISO/IEC 13818-3:1998, *Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio*.

ITU-R Recommendation BT.601-5, *Encoding parameters of digital television for studios.*

IEEE Standard Specifications for the Implementations of 8 by 8 Inverse Discrete Cosine Transform, IEEE Std 1180-1990, December 6, 1990.

IEC 60461:2001, *Time and control code for video tape recorder.*

ITU-T Recommendation H.261:1993, *Video codec for audiovisual services at p x 64 kbit/s.*

ITU-T Recommendation H.263:2000, *Enhanced reference picture selection mode.*

Withdrawn